

## Курс: Программная инженерия

- [Форма записи на проекты](#)
- [SWEBOK](#)
- [Quick overview of SWEBOK](#)

## Список проектов

### 1. Плагин vagrant для клонирования виртуальных машин

- vagrant configuration cloning plugin (VCCP)
- Задачей проекта является создание расширения для vagrant, которое бы позволило клонировать конфигурации произвольного размера для следующих провайдеров виртуализации: docker, libvirt, virtualbox
- задействованные технологии: vagrant, docker, libvirt, virtualbox
- языки разработки: ruby, bash
- <https://dev.osll.ru/projects/mdbci>, <https://github.com/OSLL/mdbci>
- **deliverables:** исходный код плагина, тесты для плагина, публикация плагина в репо плагинов для vagrant

### 2. Визуализация результатов опросов

Необходимо написать плагин, который строит графики с помощью библиотеки языка Python matplotlib (<http://matplotlib.org/>). На вход подаются таблицы формата csv, результат работы программы - файл pdf с построенными графиками.

**deliverables:** библиотека функций и консольное приложение на python 2.7 для построения кастомизируемых (настраиваемых пользователем) графиков к различным таблицам формата csv.

### 3. Автоматизация развертывания OpenStreetMaps

<https://geo2tag.atlassian.net/wiki/pages/viewpage.action?pageId=49348611>

**deliverables:** описаны в документе

### 4. ИС Кафедры: Научная работа

Приложение должно предоставлять возможность ввода/обработки/хранения/предоставление информации о:

- публикациях/научных трудах (группировка по годам/людям)
- участии (достижениях) в конференциях/конкурсах/НИОКР и т.д.
- численности аспирантов, докторантов, соискателей и стажеров

Также приложение должно обеспечивать разные уровни доступа к информации для различных пользователей.

**deliverables:** веб-приложение на языке Python2.7/Flask + MongoDB.

## 5. ИС Кафедры: Оборудование и ПО

Приложение должно предоставлять возможность ввода/обработки/хранения/предоставление информации об основных средствах кафедры, их характеристиках/о списанных технических и нетехнических средствах; а также получении информации о:

- инвентарных, заводских номерах и других идентификационных параметрах объектов;
- наличии и/или расположении некоторого технического или нетехнического средства, обладающего требуемыми параметрами;
- объектах, подлежащих списанию;
- ПК, на которых установлено определенное ПО;
- сетевых данных компьютера (сетевое имя, IP);
- списках программных, технических или др. материальных средств определенного типа, назначения, определенной фирмы и т.п.

Также приложение должно обеспечивать разные уровни доступа к информации для различных пользователей.

**deliverables:** веб-приложение на языке Python2.7/Flask + MongoDB.

## 6. Инфраструктура курса "Программирование в ядре Linux"

Главной задачей курса является освоение способов отладки и расширения ядра. Для того, чтобы сделать курс массовым, необходимо разработать технологию быстрой автоматизированной загрузки студенческих наработок в ядро Linux, запущенное в libvirt (qemu).

В процессе разработки вы получите знания в следующих областях:

- Методы модификации ядра Linux (создание и загрузка модулей ядра/драйверов/ прочих расширений)
- Методы перекомпиляции ядра
- Создание baseimage для docker
- Создание libvirt image
- Модификация libvirt image для совместимости с vagrant

**deliverables:** библиотека и консольное приложение на языке Ruby, позволяющая сгенерировать и запустить ядро Linux с пользовательскими доработками с помощью связки docker+vagrant.

## 7. Графовый фронтенд для трекеров задач

Задача: сделать веб-приложение для популярных трекеров задач (Redmine/Jira/Trello ...),

которое позволяет

- отобразить графически связи между задачами и порядок их выполнения (в виде графа)
- предоставить инструмент для графического редактирования задач (обмен местами в цепочках связей, перестановка задач, выделение критических групп задач/циклов/взаимозависимостей/недокомпозованных задач

**deliverables:** веб-приложение с регистрацией + настраиваемым подключением к одному из трекеров, возможностью редактировать и визуализировать связи между задачами.

## 8. Логистический сервис на базе Geo2Tag

<https://geo2tag.atlassian.net/wiki/pages/viewpage.action?pageId=45514805>

**deliverables:** веб-сервис для регистрации и управления автопарками + мобильное приложение водителей для передачи данных.

## 9. Планировщик туристических активностей

Приложение, которое составляет подробный план отпуска (с возможностью офлайн-работы) в незнакомом городе/стране с учетом интересов пользователя, бюджета на отпуск, местоположения отеля, параметров начала/конца отпуска, желаемого темпа, климатических условий.

**deliverables:** веб-сервис с функциями: - регистрации, - планирования, корректировки и сохранения планов отпуска.

## 10. Сбор статистики курса "Основы программирования для Linux"

Необходимо разработать расширение (blueprint) для веб-сервера проверяющей системы курса "Основы программирования для Linux", которое бы позволяло решить следующие задачи:

- потоковый (без записи всего лога в память) анализ лога, выявление отдельных событий и их длительности (опционально) в рамках заданного временного интервала
  - поступление заявки
  - обработка заявки
  - внутренние ошибки системы
  - произвольные события (задана подстрока начала и конца события)
  - вычисление статистических моментов от заданной выборки (максимальное, минимальное, среднее, частота события)
- отображение и визуализация данных на веб-странице
  - страница имеет GET параметры
    - временной интервал (может быть открытым)
    - группировка значений (по часам, дням, неделям)
    - отображаемое событие или события
  - отображение в виде временной зависимости
    - визуализация с помощью библиотеки <https://developers.google.com/chart/>
    - на графике необходимо отображать статистические моменты

**deliverables:** веб-сервис на Python2.7/Flask, содержащий веб-страницы для графического отображения статистики за заданный период с заданными настройками.

## 11. Сервис запуска docker-контейнеров по требованию

Разработать веб-сервис на Python2.7/Flask для запуска docker-контейнеров по требованию с помощью vagrant. Требования:

- REST интерфейсы
  - запуск/status/остановка контейнера
  - получение списка запущенных контейнеров
  - создание/редактирование/отображение Vagrantfile
- веб-интерфейс для управления запущенными контейнерами:
  - требует авторизации
  - отображение таблицы с запущенными контейнерами и ссылками на их конфигурации
  - элементы управления для работы с контейнерами и Vagrantfile

**deliverables:** веб-сервис на Python2.7/Flask с поддержкой авторизации и содержащий веб-страницу для управления контейнерами и конфигурациями.

## 12. Remote AVR development IDE

Разработать решение для удаленной загрузки и отладки программ в микроконтроллерах AVR. К серверу подключается несколько микроконтроллеров через программаторы отладчики по интерфейсу JTAG, доступные для пользователей. Каждый пользователь, используя клиентское приложение, может забронировать себе некоторый интервал времени, в который ему будет доступен один из микроконтроллеров. Клиентским приложением является Arduino IDE с добавленным функционалом для дистанционной отладки написанных в ней программ с использованием отладчика GDB.

**deliverables:** Модифицированная среда разработки Arduino IDE, сервис управление сервером отладки, сервис резервирования контроллеров для удаленного доступа.

## 13. ИС Кафедры: индивидуальные планы преподавателей

Требуется разработать компоненты информационной системы кафедры, обеспечивающие управление индивидуальными планами преподавателей

- Планирование учебной нагрузки
- Переподготовка и повышение квалификации
- Участие в мероприятиях, конференциях,
- Руководство НИР
- Генерация отчетов
- ... и другие параметры индивидуального плана

## 14. ИС Кафедры: учет студентов

Требуется разработать компоненты информационной системы кафедры, обеспечивающие:

- Управление студенческими данными
- Импорт/экспорт данных
- Рейтинги по предметам, курсовые, практики.
- Генерация отчетов.

From:

<http://se.moevm.info/> - **se.moevm.info**

Permanent link:

<http://se.moevm.info/doku.php/start?rev=1473415182>

Last update: **2022/12/10 09:08**

